Theoretical Computer Science 64 (1989) 55–66
North-Holland

E-CCC: BETWEEN CCC AND TOPOS

Hiroyuki SATO

Department of Information Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

Communicated by M. Nivat
Received March 1987
Revised January 1988

Abstract. The cartesian closed category (ccc) and topos differ in both descriptive power and executability. The ccc cannot express the concept of subtypes, while it has the executable structure as a model of typed λ-calculus. On the other hand, the topos has strong expressive power of subtypes, although it does not in general have a good correspondence to any computation system. This paper introduces the structure of e-ccc, as an intermediate of the ccc and topos. The e-ccc has the correspondence to the λ-calculus based on an extended abstract data type theory and thus can be considered to be executable. Moreover, relations between e-ccc and ccc or topos are discussed. In particular, the topos is proved to be a specially-structured e-ccc.

1. Introduction

Categorical methods are of current interest for the description and analysis of computation systems. In particular, the cartesian closed category (ccc for short) is studied as a computation system. It is proved to have a correspondence to λ-calculus [1]. Intensive study has been performed from the above point of view [3, 4]. Topos is also intensively studied in view of the relation to higher order intuitionistic logic [2, 3]. Topos can be considered as the specially-structured ccc.

Both ccc and topos have advantages and disadvantages. The ccc has good correspondence to the typed λ-calculus, one of the standard computation systems. It has, however, the rather weak expressive power of properties of types. On the other hand, topos has the strong descriptive power of subtypes, or properties of types, although it does not have a good correspondence to any computation system. It does correspond to the higher order intuitionistic type theory, however, the type theory is far from executing, or computing. To fill up these gaps has a particular significance in data type theory.

In data type theory, the method of solving the problem of subtypes determines the descriptive power of the theory. It is one of the key problems of data type theory in computer science. Since computer science is keenly interested in executability, or computability, the strong expressive power of subtypes is not necessarily desirable. For example, consider the case of topos which regards subtypes as formulas. The formula $p(x)$ corresponding to a certain subtype does not have any meaning in

computer science, if for some term t, neither $\vdash p(t)$ nor $\nvdash p(t)$ can be proved. This problem requires a careful solution.

This paper introduces the structure of e-ccc. It allows the construction of equalizers for a certain class of arrows. Because equalizers are monic, their construction can be regarded as the description of subtypes for a restricted class.

In this paper, Section 3 defines e-ccc, and Section 4 and Section 5 prove that e-ccc can be considered to be ADT^e-λ, the λ-calculus constructed on an extension of the abstract data type theory, which in fact can be considered to be executable. This corresponds to the relation of ccc to typed λ-calculus, an executable system. The e-ccc, in this meaning, can be considered to be computable. In Section 6, topos is described as the special case of e-ccc. The executability of topos is also discussed.

2. Preliminaries

This section gives definitions and notations used in later sections.

Definition 2.1 (*Cartesian category*). A category C is *cartesian* if

(i) C has the terminal object,

(ii) if a and b are objects of C, then the Cartesian product $a \times b$ is also an object of C.

Definition 2.2 (*Cartesian closed category*). A cartesian category C is *cartesian closed* if for every object a in C, the right adjoint $a^{(-)}$ (*exponential*) to $a \times (-)$ can be defined. We also call this category ccc for short.

Lambek and Scott [3] give the equational presentation of ccc. The presentation below is that of [3].

Definition 2.3 (*Equational presentation of ccc*)

(E1) For $f: A \to B$, $f \circ \mathrm{id}_A = f$, $\mathrm{id}_B \circ f = f$.
For $f: A \to B$, $g: B \to C$ and $h: C \to D$,

$$(h \circ g) \circ f = h \circ (g \circ f).$$

(E2) For all $f: A \to 1$, $f = !_A$.

(E3) The arrows $\pi_{A,B}: A \times B \to A$ and $\pi'_{A,B}: A \times B \to B$ are defined for every pair of objects A and B and satisfy

$$\pi_{A,B}\langle f, g\rangle = f, \qquad \pi'_{A,B}\langle f, g\rangle = g,$$

$$\langle \pi_{A,B} \circ h, \pi'_{A,B} \circ h\rangle = h.$$

(E4) For every pair of objects A and B, the arrow $\mathrm{ev}_{A,B}: A^B \times B \to A$ is defined. Moreover, for every arrow $h: C \times B \to A$, the arrow $h^*: C \to A^B$ is defined, and the following equations hold:

$$\mathrm{ev}_{A,B} \circ \langle h^* \circ \pi_{C,B}, \pi'_{C,B} \rangle = h,$$

$$(\mathrm{ev}_{A,B} \circ \langle k \circ \pi_{C,B}, \pi'_{C,B} \rangle)^* = k.$$

Using this presentation, Lambek and Scott [3] showed the one-to-one correspondence between ccc and typed λ-calculus.

Definition 2.4 (*Topos*). *Topos* is the cartesian closed category in which the subobject functor is representable. The representing object is denoted by Ω. In other words, there is an object Ω and an arrow $\top: 1 \to \Omega$ which satisfy the following conditions.

(i) For every mono m, there is a uniquely determined arrow χ which makes the diagram below pullback. We denote this arrow χ as char(m).

$$\begin{array}{ccc} D & \overset{m}{\rightarrowtail} & E \\ \downarrow & PB & \downarrow \mathrm{char}(m) \\ 1 & \overset{\top}{\longrightarrow} & \Omega \end{array}$$

(ii) Conversely, for every arrow $\chi: A \to \Omega$, there corresponds a mono m which makes the above diagram pullback. This correspondence is unique up to isomorphism.

Topos is proved to have a correspondence to higher-order intuitionistic type theory [2, 3].

3. E-ccc

This section introduces e-ccc, an extension of ccc. The e-ccc allows the construction of equalizers for a restricted class of arrows. The concept of subtypes is expressed using this construction of equalizers. In e-ccc, the moderate description of subtypes is possible.

Definition 3.1 (*e-property*). Given a cartesian category C and a subclass E of objects of C which satisfy (*) and (**), below, the pair $\langle C, E \rangle$ is called a *category C with e-property E.*

(*) The terminal object 1 is in E.

(**) If both a and b are in E, then $a \times b$ is also in E.

In the category C with e-property E, we say that an object a has the e-property if a is in E. If E is clear from context, we drop E and simply say "category C with e-property".

Definition 3.2 (*e-cc*). A category C with e-property E is *e-cc* if it satisfies the following conditions.

(1) Let two objects a and b be given, with b having the e-property. For an arbitrary pair of arrows $f, g: a \to b$, the diagram $a \rightrightarrows^{f}_{g} b$ has the equalizer $\mathrm{inc}_{f,g}: \mathrm{Eq}(f, g) \rightarrowtail a$.

(2) In the above diagram, if a has the e-property, then $\mathrm{Eq}(f, g)$ also has the e-property.

(3) $\mathrm{Eq}(f, g) = \mathrm{Eq}(g, f)$.

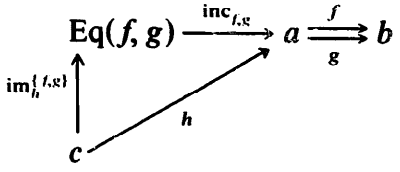
(4) $\mathrm{Eq}(\langle f, g\rangle, \langle f', g'\rangle) = \mathrm{Eq}(\langle g, f\rangle, \langle g', f'\rangle)$.

(5) $\mathrm{Eq}(\langle f, g\rangle, \langle f', g'\rangle) = \mathrm{Eq}(\langle f, g'\rangle, \langle f', g\rangle)$.

The conditions (3), (4) and (5) are rather technical than essential.

Definition 3.3 (*e-ccc*). An e-cc C is *e-ccc* if it is cartesian closed.

Definition 3.4 (*Equational presentation of e-ccc*). Just as ccc, e-ccc has the equational presentation. Let $\mathrm{Eq}(f, g)$ be defined. From the definition of equalizer, for $h: c \to a$ such that $f \circ h = g \circ h$, we have a unique arrow j such that $\mathrm{inc}_{f,g} \circ j = h$. We denote it by $\mathrm{im}_h^{\{f,g\}}$. We also denote it by im_h if there is no ambiguity.



In addition to the presentation from (E1) to (E4) of the ccc,

(E5.1) $\mathrm{inc}_{f,g} \circ \mathrm{im}_h^{\{f,g\}} = h$

(E5.2) $\mathrm{im}^{\{f,g\}}_{\mathrm{inc}_{f,g} \circ k} = k$.

The proposition below shows some useful equations.

Proposition 3.5

(1) $\mathrm{im}_h^{\{f,g\}} \circ k = \mathrm{im}^{\{f,g\}}_{h \circ k}$.

(2) $\mathrm{im}_{\mathrm{inc}_{f,g}} = \mathrm{id}_{\mathrm{Eq}(f,g)}$.

(3) $\mathrm{inc}_{f,f}$ *is an isomorphism.*

Proof.

(1) $\mathrm{im}_h^{\{f,g\}} \circ k = \mathrm{im}^{\{f,g\}}_{\mathrm{inc}_{f,g} \circ \mathrm{im}_h \circ k}$ (E5.2)

$= \mathrm{im}^{\{f,g\}}_{h \circ k}$. (E5.1)

(2) $\mathrm{im}_{\mathrm{inc}_{f,g}} = \mathrm{im}_{\mathrm{inc}_{f,g} \circ \mathrm{id}} = \mathrm{id}$. (E5.2)

(3) The arrow $\mathrm{im}_{\mathrm{id}}$ is the inverse of $\mathrm{inc}_{f,f}$ because

$\mathrm{inc}_{f,f} \circ \mathrm{im}_{\mathrm{id}} = \mathrm{id}$ (E5.1)

$\mathrm{im}_{\mathrm{id}} \circ \mathrm{inc}_{f,f} = \mathrm{im}_{\mathrm{inc}_{f,f}} = \mathrm{id}$. (1), (2) □

The result (3) of the above proposition indicates that $\mathrm{inc}_{f,f}$ can be regarded as an identity. We therefore require the following condition (added axiom to Definition 3.2).

(6) Let $f: a \to b$ be an arrow with b having the e-property. Then

$$\mathrm{Eq}(f,f) = a, \qquad \mathrm{inc}_{f,f} = \mathrm{id}_a.$$

Definition 3.6 (E-CCC). Let **E-CCC** be the category of e-ccc, whose objects are e-ccc's and whose arrows are functors preserving cartesian products and exponentials added with e-property and related equalizers.

Remark 3.7 (*Expressive power of subtypes*). In categorical terms, the concept of subtypes corresponds to that of mono. In e-ccc, we can structurally and concretely construct a mono as an equalizer. In the subsequent sections, e-ccc is proved to correspond to an extended ADT, the abstract data type theory which can restrictedly express the concept of subtypes. Using equalizers and terms of the ADT, we can represent decidable subtypes.

The subclass E of e-ccc C determines its representative power of the e-ccc. If E consists of only terminal object, C is just a ccc. Section 6 discusses the relation of e-ccc and topos in view of e-property.

4. ADT^e-λ

This section and the next section investigate the correspondence between the computation system ADT^e-λ and the category e-ccc.

4.1. Equational logic with $\to$

Definition 4.1.1 (*ADT*). The pair $\langle \Sigma, A \rangle$ satisfying the following conditions is called ADT.

(Σ) A signature Σ consists of the class of "types" and "terms" satisfying the following conditions:

Types: 1 is a type. If P and Q are types, then $P \times Q$ is also a type.

Terms: (i) Countably many variables are assigned to each type.

(ii) There are fixed a set of function symbols including $\langle -,- \rangle$, π and π'.

(iii) Each term is assigned to a type. In particular, if a term t is of type P and a term s of type Q, then $\langle t, s \rangle$ is of type $P \times Q$. Moreover, if a term r is of type $P \times Q$, then $\pi(r)$ is of type P and $\pi'(r)$ of type Q. The term $*$ is of type 1.

(A) The class of axioms which is a subclass of formulas of EL(Σ). The class of formulas EL(Σ) is defined as follows.

Definition 4.1.2 (*EL*(Σ)). Given a signature Σ, EL(Σ) is defined in the following way.

(i) For every type A, the equality $=_A$ is defined. We sometimes drop the subscript if the type A is clear from context.

(ii) Formulas in EL(Σ) are:

If a and b are terms of type A in Σ, then $a =_A b$ is a formula.

If p and q are formulas in EL(Σ), then p & q is also a formula.

(iii) There are axioms and inference rules of &, equality, $\langle -,- \rangle$, π and π'.

We write $\vdash p$ meaning that the formula p is provable in EL(Σ). We drop Σ and simply write EL, if Σ is clear from context.

Definition 4.2 (*EL*$^{\rightarrow}$(Σ)). The equational logic with implication ($\rightarrow$) (EL$^{\rightarrow}$(Σ)) is an extension of EL(Σ) added with the following properties:

(i) The symbol $\rightarrow$ is added to the set of logical symbols.

(ii) Formulas in EL(Σ) are also formulas in EL$^{\rightarrow}$(Σ). Given a formula p in EL(Σ) and q in EL$^{\rightarrow}$(Σ), $p \rightarrow q$ is a formula in EL$^{\rightarrow}$(Σ).

(iii) The axioms of implication ($\rightarrow$) are added.

In fact, we have a logically equivalent presentation of EL$^{\rightarrow}$(Σ) (ii).

Lemma 4.3. *We have an equivalence by changing* (ii) *of the above definition as following*:

(ii)′ *Given a formula p in* EL(Σ) *and q in* EL(Σ), $p \rightarrow q$ *is a formula in* EL$^{\rightarrow}$(Σ).

Proof. By the induction on the construction of term q.

(i) *Basis*: If q is a term of EL, nothing remains to be proved.

(ii) *Induction step*: Let q have the form $q' \rightarrow q''$, where q' is a formula of EL and q'' of EL$^{\rightarrow}$. From the induction hypothesis, q'' is equivalent to the form $p \rightarrow p'$, with both p and p' in EL. Then $q \leftrightarrow (q' \rightarrow (p \rightarrow p'))$. It is equivalent to $(q' \,\&\, p) \rightarrow p'$, in the form of (ii)′. □

Hereafter, we use (ii)′ instead of (ii). We also write simply EL$^{\rightarrow}$ for EL$^{\rightarrow}$(Σ), if Σ is clear from context.

Notation 4.4. We say that a formula $p(x)$ is of type A if $p(x)$ contains only one variable x and its type is A.

Definition 4.5 (*ADT*^{e}). The pair $\langle \Sigma, A \rangle$ satisfying the following conditions together with those for ADT is called ADT^{e}.

(Σ) The signature Σ is defined simultaneously in terms of *types* and *terms*.

Types: For every formula $p(x)$ in EL(Σ) containing one variable, $\overline{p(x)}$ is defined and is a type.

Terms: New constructors of terms are added in the following way:

(i) Let $p(x)$ be a formula of type A, and a a term of type $\overline{p(x)}$. The term $\mathrm{inc}_A^{p(x)}(a)$ is defined and of type A.

(ii) Let b be a term of type B such that $\vdash p(b)$. In this case, the term $\mathrm{im}^{A}_{p(x)}(b)$ is defined and of type $\overline{p(x)}$.

(iii) The constructors inc and im satisfy:

$$\mathrm{inc}^{p(x)}_{A}(\mathrm{im}^{A}_{p(x)}(b)) = b, \qquad \mathrm{im}^{A}_{p(x)}(\mathrm{inc}^{p(x)}_{A}(a)) = a.$$

(A) A is the class of axioms written in EL_0 in which for every term a of type $\overline{p(x)}$, $p(a)$ holds.

The constructors inc and im change the type of terms. We drop their subscripts and superscripts and simply write inc and im if there is no ambiguity.

Example 4.6. Consider the ADT $\langle \Sigma, A\rangle$ defined as

$\Sigma \equiv N$ as type,

0, suc, pred as term constructors,

$A \equiv \{\mathrm{pred}(0) = 0, \mathrm{pred}(\mathrm{suc}(x)) = x\}$.

We can construct the $\mathrm{ADT}^{e}\langle \Sigma', A'\rangle$ from $\mathrm{ADT}\langle \Sigma, A\rangle$. In Σ', we also have for example types corresponding to formulas $\mathrm{suc}(\mathrm{pred}(x)) = x$, $\mathrm{suc}(\mathrm{suc}(\mathrm{pred}(\mathrm{pred}(x)))) = x$, etc. Informally, the former corresponds to the subtype $\{x \mid x > 0\}$ and the latter to $\{x \mid x > 1\}$.

Definition 4.7 (*ADT'*). The pair $\langle \Sigma, A\rangle$ is ADT' if it is ADT except that formulas of A are those written in $\mathrm{EL}^{\rightarrow}(\Sigma)$.

ADT' are today widely used as an expedient in order to express the properties which cannot be described solely in ADT. In fact, ADT' can be expressed in ADT^{e}. For example, consider the axiom $p(x) \rightarrow q(x)$ in ADT'. In ADT^{e}, it can be expressed as $q(\mathrm{inc}(x))$ for a variable x of type $\overline{p(x)}$.

Remark 4.8 (*Functions on subtypes*). By restricting the domain of functions, we obtain some useful properties. For example, in the above example, suc has the inverse pred only if its domain satisfies $\mathrm{suc}(\mathrm{pred}(x)) = x$. In ADT^{e}, such properties are easily expressed. This is one of the advantages of ADT^{e}.

Definition 4.9 (*ADT^{e}-λ: λ-calculus on ADT^{e}*). Given an $\mathrm{ADT}^{e}\langle \Sigma, A\rangle$, we define its associated ADT^{e}-λ as follows:

Types:

(i) Types of ADT^{e} are also types of ADT^{e}-λ.

(ii) If A and B are types, then $A \times B$ and A^{B} are also types. In other words, the class of types is cartesian closed.

(iii) Equalities are defined only on types of ADT^{e}.

(iv) For every formula $p(x)$ in EL defined on terms described below, $\overline{p(x)}$ is defined and is a type.

Terms:

(i) Terms of ADT^e are terms of ADT^e-λ.

(ii) Countably many variables are assigned to each type.

(iii) There are constructors in λ-calculus. We do not specify them here. The reader should consult [3].

(iv) There are added new constructors inc and im.

5. E-CCC and ADT^e-λ

This section proves that ADT^e-λ and e-ccc are equivalent concepts. The discussion of this section is based on the corresponding part of [3], so readers are assumed to be grounded in its results and notations.

5.1. (L) the internal language

Given an e-ccc C, we consider its internal language $L(C)$, the ADT^e-λ by simultaneous induction of types and terms as follows.

Definition 5.1 ($L(C)$).

Types: Types of $L(C)$ are objects of C, with

(i) Eq(f, g) is transferred to $\overline{fx = gx}$. By the axiom of pair of arrows, it is easily checked that Eq$(\langle f, g\rangle, \langle f', g'\rangle)$ corresponds to $\overline{fx = f'x \;\&\; gx = g'x}$.

(ii) Products of C are transferred to products of types.

(iii) Exponentials of C are transferred to exponentials of types.

(iv) 1 in C is transferred to Type 1.

Terms: Terms of type A are those polynomial expressions $\phi(x): 1 \to A$ in the indeterminate $x: 1 \to A_x$ obtained from variables, and constants by the term constructing operations:

$$\frac{a: 1 \to A \qquad b: 1 \to B}{\langle a, b\rangle: 1 \to A \times B} \qquad \frac{a: 1 \to A}{fa: 1 \to B}$$

where $f: A \to B$.

$$\frac{\phi(x): 1 \to B}{\lambda x \in A\,.\,\phi(x): 1 \to B^A}$$

where $\lambda x \in A\,.\,\phi(x)$ is defined as that of [3].

$$\frac{fa = ga}{\mathrm{im}^A_{fx=gx}(a): 1 \to \overline{fx = gx}}$$

where A is the type of a. Moreover, we identify the constructor $\mathrm{inc}_{f,g}$ with $\mathrm{inc}_A^{fx=gx}$, where A is the domain of f and g.

ADT^e: The ADT^e is defined as the cartesian subcategory whose objects have the e-property.

The equality of terms is defined as the equality of arrows in C.

Definition 5.2 (ADT^e-Λ). Given two ADT^e-λ L and L', Φ sending types and terms of L to those of L' is called *translation* if it preserves type constructors (products, exponentials, equalizers, etc), term constructors (pairing, projection, etc), axioms and e-property.

We obtain a category **ADT^e-Λ** whose objects are ADT^e-λ and arrow translations.

Proposition 5.3. *Let $L(C)$ be the internal language of C, and, for any translation $F: A \to A'$, let $L(F)$ be defined by*

$$L(F)(A) \equiv F(A) \qquad L(F)(x_i) \equiv x_i', \qquad L(F)(\phi(x)) \equiv F(\phi(x)).$$

Then L is a functor from **E-CCC** *to* **ADT^e-Λ.**

5.2. The e-ccc generated by an ADT^e-λ

In order to show that the functor L is an equivalence of categories, we construct the functor E in the opposite direction.

Definition 5.4 (E). Given an ADT^e-λ L, we construct an e-ccc $E(L)$ in the following way.

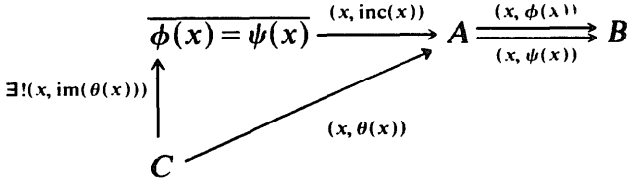
Objects: Objects in $E(L)$ are types in L.

Arrows: Arrows $A \to B$ of $E(L)$ are (equivalence classes of) pairs $(x \in A, \phi(x))$, with x a variable of type A and $\phi(x)$ a term of type B with no free variables other than x.

E-property: An object A has the e-property if $=_A$ is defined for the type seen from ADT^e-λ.

Lemma. *The object $\overline{\phi(x) = \psi(x)}$ is the equalizer of the arrows $(x, \phi(x))$ and $(x, \psi(x))$.*

Proof. Consider the following diagram:



Suppose $(x, \phi(x)) \circ (x, \vartheta(x)) = (x, \psi(x)) \circ (x, \theta(x))$. This implies that $\phi(\theta(x)) = \psi(\theta(x))$. Then, there is a unique arrow $(x, \mathrm{im}(\theta(x)))$ from C to $\overline{\phi(x) = \psi(x)}$ which make the above diagram commute. □

By the above lemma, we identify the object $\overline{\phi(x)=\psi(x)}$ with $\mathrm{Eq}((x,\phi(x)),(x,\psi(x)))$. Thus, we define

$$\mathrm{inc}_{(x,\phi(x)),(x,\psi(x))} \equiv (z \in \mathrm{Eq}((x,\phi(x)),(x,\psi(x))), \mathrm{inc}_A^{\overline{\phi(x)=\psi(x)}}(z)),$$

where A is the type of x.

$$\mathrm{im}_h^{\{(x,\phi(x)),(x,\psi(x))\}} \equiv (z \in C, \mathrm{im}(z)),$$

where $h: C \to A$.

The type corresponding to &-conjuncted formulas are interpreted as an equalizer of a pair of arrows. For example, a formula $\phi(x)=\psi(x)\ \&\ \phi'(x)=\psi'(x)$ corresponds to

$$(x, \langle \phi(x), \phi'(x)\rangle) = (x, \langle \psi(x), \psi'(x)\rangle).$$

As for cartesian products and exponentials in e-ccc, we also identify them with those of ADT^e-λ. Thus,

$$!_A \equiv (x \in A, *), \qquad \pi_{A,B} \equiv (z \in A \times B, \pi z), \qquad \pi'_{A,B} \equiv (z \in A \times B, \pi' z),$$

$$\langle (z, \phi(z)\rangle, (z, \psi(z))\rangle \equiv (z, \langle \phi(z), \psi(z)\rangle),$$

$$(z, \chi(z))^* \equiv (x, \lambda y.\, \chi(\langle x, y\rangle)),$$

$$\mathrm{ev} \equiv (y, \mathrm{ev}(\langle \pi(y), \pi'(y)\rangle)).$$

We can easily extend E to the functor from **ADT^e-Λ** to **E-CCC.**

Theorem 5.5 (*Equivalence of L and E*).

(i) $E \circ L \sim \mathrm{id}$.

(ii) $L \circ E \sim \mathrm{id}$.

Proof. (i): First, we prove the equivalence of the e-ccc C and $E(L(C))$. Equivalence between objects is clear from the construction of L and E. An arrow $B \to C$ in $EL(C)$ has the form $(y, \phi(y))$.

Claim. *In the above condition, there is a unique arrow $g: B \to C$ such that $gy = \phi(y)$.*

Proof. By induction on the construction of $\phi(y)$. In fact, this is the functional completeness of e-ccc. The functional completeness holds in ccc, so we only have to prove the case below.

In case $\phi(y) = \mathrm{im}_{\theta(y)}^{\psi(x)=\psi'(x)}$, let f, g, and h be arrows such that $fx = \psi(x)$, $gx = \psi'(x)$, and $hx = \theta(x)$. Then, $fh(x) = gh(x)$, so $fh = gh$. Therefore, $\mathrm{im}_h^{\{f,g\}}$ can be defined and $\mathrm{im}_{hx} = \mathrm{im}_h \circ x$. □

By the above claim, we have the one-to-one correspondence between arrows of $EL(C)$ and of C by sending $(x, \phi(x))$ to such g in the claim.

(ii): For any ADT^e-λ L, we prove the equivalence between $LE(L)$ and L. It is straightforward as for types and equality. The term $\phi(x)$ in L corresponds to $(z \in 1, \phi(z))$ in $LE(L)$. The correspondence is onto and one-to-one, since we can send the latter to the former by $(z \in 1, \phi(z))(*) = \phi(*)$. □

6. E-CCC and topos

This section shows that topos can be considered as a special case of e-ccc.

Proposition 6.1. *Every subobject of topos is an equalizer of some pair of arrows. In other words, every mono is regular.*

Proof. Let m be a mono. The arrow m is the equalizer of char(m) and $\mathsf{T} \circ !$. □

The above proposition shows that topos can be considered as higher-order ADT^e-λ, provided that Ω has the e-property and that the functors Sub and Hom(–, Ω) are isomorphic to each other. The latter condition informally means that in that category, we have and only have a "canonical" subobject. For example, if the topos is skeletal, the condition is satisfied.

6.1. Executability of topos

The executability of ADT^e-λ of e-ccc relies on whether equalizers can be constructively obtained. In other words, it depends on whether the equality of the ADT^e-λ is decidable or not. Therefore, thinking that the usual ADT^e-λ is computable, e-ccc is sufficiently considered as executable. We do not require that the e-property should be cartesian closed. It reflects the fact that the extensional equality of functions is generally not decidable.

Consider from this viewpoint the executability of topos. The equality relation on Ω is $\leftrightarrow$. In general, we do not expect that the relation $\leftrightarrow$ is decidable. In this meaning, topos cannot be executable. If we construct the execution system based on topos, it will automatically include a certain kind of prover.

7. Conclusion and future research

An extension e-ccc, of ccc is introduced to increase the expressive power of ccc. It is proved to have an equivalent relation to the ADT^e-λ, the λ-calculus based on ADT^e in the meaning of Theorem 5.5. In the usual sense, ADT^e-λ can be considered as a computable system. Thus e-ccc can also be considered as computable extension of ccc. Its relation with topos is also discussed. The executability of topos, or higher-order intuitionistic type theory, is analyzed using e-ccc.

Today, much contribution is made to implement the category theory as computing system. The ccc is one of the most successful, because it corresponds to typed λ-calculus. The e-ccc is hopeful from the above viewpoint. In future, in order to implement the subclass of the data type theory, e-ccc can be one candidate, because it is also regarded as a data type theory. If its implementation is processed by the restriction of topos structure, e-ccc is again hopeful because it is one of the restricted structures of topos.

Acknowledgment

The author is specially grateful to Professor Nobuo Yoneda for his relevant advice and encouragement.

References

[1] H.P. Barendregt, *The Lambda Calculus. Its Syntax and Semantics* (North-Holland, Amsterdam, 1981).
[2] P. Fourman, The logic of topoi, in: J. Barwise, ed., *Handbook of Mathematical Logic* (North-Holland, Amsterdam, 1977) 1053–1090.
[3] J. Lambek and P. Scott, *Introduction to Higher Order Categorical Logic* (Cambridge University Press, London, 1986).
[4] H. Yokouchi, Cartesian closed structures in models of lambda calculus, Res. Rept. C-51, Dept. of Information Science, Tokyo Institute of Technology, 1983.